Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

**Отчёт**

по предмету «Проектирование интернет систем»

Лабораторная работа №7

«Объектно-ориентированное моделирование. Диаграммы поведения UML»

Студент: Фамилия И. О.

ФИТ 4 курс 6 группа

Преподаватель: Якубенко К. Д.

Минск 2024

**Введение**

Цель лабораторной работы по теме "Объектно-ориентированное моделирование. UML-диаграммы поведения" заключается в углубленном изучении методологии объектно-ориентированного моделирования (ООМ) с использованием UML (Unified Modeling Language) для создания и анализа диаграмм поведения программных систем. Объектно-ориентированное моделирование представляет собой ключевой подход в программировании, направленный на описание и разработку сложных систем в виде взаимодействующих объектов, что позволяет создавать устойчивую, гибкую и масштабируемую архитектуру программных решений. Работа с UML-диаграммами поведения помогает лучше понять функциональные требования к системе и то, как различные компоненты взаимодействуют между собой.

Одной из задач лабораторной работы является ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения, такими как инкапсуляция, наследование и полиморфизм, которые лежат в основе объектно-ориентированного подхода. Эти принципы позволяют более эффективно управлять сложностью разработки программных систем, что способствует созданию структурированных и понятных проектов. В ходе работы изучаются методы проектирования системы в виде объектов и их взаимосвязей, что играет важную роль в современном программировании.

В ходе лабораторной работы внимание уделяется моделированию динамики поведения объектов и их взаимодействия. Рассматривается, как объекты системы обмениваются сообщениями, изменяют свое состояние и реагируют на события. Описание динамических взаимодействий позволяет лучше понять, как поведение объектов изменяется во времени, что способствует более точному отражению требований к системе и ее адаптации к будущим изменениям.

Еще одной целью работы является овладение навыками анализа и корректировки UML-моделей для обеспечения их соответствия реальным требованиям информационных систем. UML позволяет анализировать как статическую, так и динамическую структуру системы, выявляя возможные ошибки на этапе проектирования. В процессе работы формируется понимание того, как выявлять узкие места в системе, проверять корректность поведения объектов и их взаимодействий, а также предлагать возможные улучшения и оптимизации.

Кроме того, выполнение лабораторной работы направлено на развитие практических навыков использования UML как универсального инструмента для документирования и анализа информационных систем. Осваиваются программные средства и инструменты, позволяющие автоматизировать процесс создания UML-диаграмм и интегрировать UML в процесс разработки программного обеспечения. Это необходимо как для проектирования новых систем, так и для анализа и поддержки существующих решений, что делает UML важным инструментом в разработке программного обеспечения.

1. **Описание программных средств**

Draw.io (или diagrams.net) — это мощный и удобный инструмент для создания разнообразных диаграмм и схем, который завоевал популярность благодаря своей функциональности и бесплатной основе. Одной из главных особенностей платформы является простота в использовании, что делает её доступной как для новичков, так и для профессионалов. Благодаря интуитивно понятному интерфейсу, пользователи могут быстро создавать различные типы диаграмм, такие как блок-схемы, диаграммы UML, сетевые диаграммы, карты процессов, модели IDEF0 и организационные структуры.

Инструмент также предлагает обширную библиотеку готовых шаблонов и элементов для создания визуализаций. Пользователи могут выбирать из множества стандартных фигур и иконок, что значительно ускоряет процесс построения диаграмм.

Платформа поддерживает множество форматов для экспорта, включая PNG, JPG, SVG и PDF, что делает ее универсальной для различных задач, таких как представление проектов, печать или размещение в электронных документах. Более того, для будущего редактирования диаграммы можно сохранить в собственном формате XML, что позволяет пользователям возвращаться к проектам в любое время для их доработки. Draw.io также интегрируется с популярными облачными сервисами, такими как Google Drive, Dropbox и OneDrive, что упрощает процесс хранения, редактирования и обмена диаграммами, а также обеспечивает доступ к проектам с любого устройства и совместную работу в реальном времени.

Для командной работы draw.io предлагает инструменты для совместного редактирования диаграмм, позволяя нескольким пользователям одновременно вносить изменения в схемы и делиться комментариями. Дополнительным плюсом является возможность работы в оффлайн-режиме через настольное приложение, что позволяет создавать и редактировать схемы даже без доступа к интернету.

Платформа также предлагает интеграцию с различными корпоративными системами и популярными платформами для управления проектами, такими как Atlassian Confluence и Jira. Это позволяет использовать draw.io для визуализации данных и процессов в рамках более широких решений для управления проектами. Для пользователей, которые уже работают с другими инструментами для диаграмм, draw.io предлагает импорт и экспорт данных, что делает его совместимым с такими сервисами, как Lucidchart и Microsoft Visio, позволяя легко переходить между различными инструментами.

Таким образом, draw.io сочетает в себе простоту использования, гибкость работы с различными форматами и возможность интеграции с облачными сервисами, что делает его одним из самых популярных инструментов для создания диаграмм и визуализации данных. Платформа является идеальным решением как для индивидуальных пользователей, так и для команд, работающих над совместными проектами, предлагая все необходимые инструменты для эффективной работы.

1. **Описание практического задания**

В рамках проектирования платформы для обучения и контроля знаний по математике использование диаграмм поведения является ключевым аспектом для визуализации процессов и взаимодействий, что способствует более ясному пониманию функционирования системы.

На рисунке 2.1 представлена диаграмма последовательности для бизнес-функции «Создание онлайн-занятия в календаре».

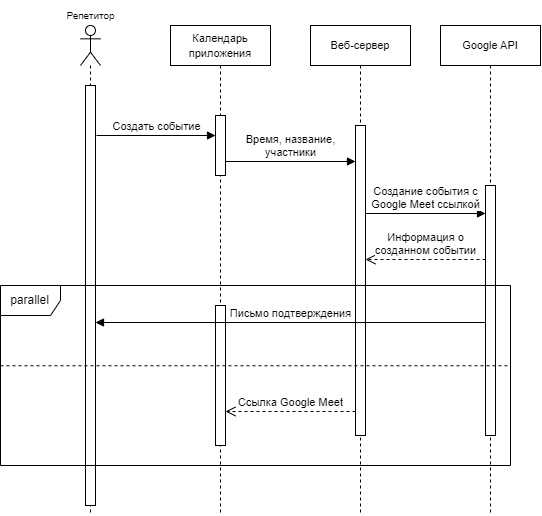


Рисунок 2.1 – Диаграмма последовательности

Основные элементы:

* репетитор – актор, который инициирует создание события;
* календарь приложения – компонент, отвечающий за создание события и передачу информации;
* веб-сервер – промежуточный сервер для обработки запроса и взаимодействия с Google API;
* Google API – интерфейс для создания события в Google Meet и получения ссылки на встречу.

Связи между элементами:

* сообщения**:** используются для передачи данных между элементами. «Создать событие» – от репетитора к Календарю приложения. «Время, название, участники» – от Календаря приложения к Веб-серверу. «Создание события с Google Meet ссылкой» – от Веб-сервера к Google API.
* «Письмо подтверждения» и «Ссылка Google Meet» – передаются обратно репетитору через цепочку обратных сообщений;
* параллельное выполнение: отображено блоком parallel, что означает одновременную отправку письма подтверждения и отображение ссылки на Google Meet в календаре приложения.

На рисунке 2.2 представлена диаграмма состояний для бизнес-функции «Создание курса».

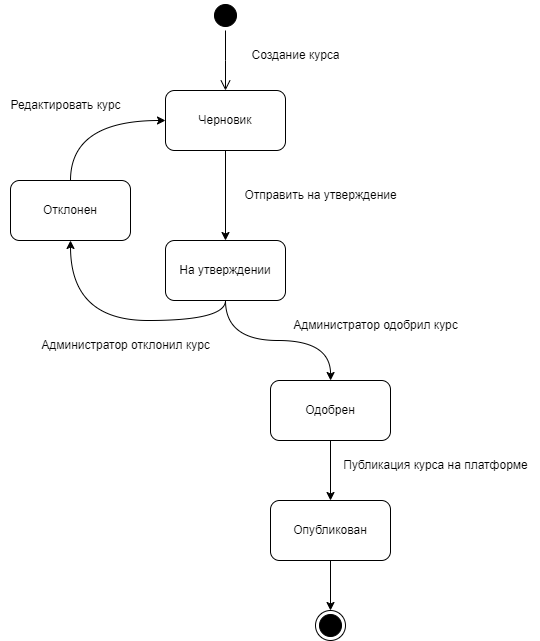


Рисунок 2.2 – Диаграмма состояний

Основные элементы:

* черновик – состояние после создания курса;
* наутверждении – состояние после отправки на утверждение;
* отклонен – курс отклонен администратором;
* одобрен – курс одобрен администратором;
* опубликован – курс опубликован на платформе.

Связи между элементами:

* переход от «Черновика» к «На утверждении» происходит при отправке курса на проверку;
* от «На утверждении**»** возможны переходы к состояниям «Отклонен» или «Одобрен» в зависимости от решения администратора;
* переход от «Отклонен» к «Черновик» при повторном редактировании курса;
* переход от «Одобрен» к «Опубликован» происходит после публикации курса на платформе.

Таким образом, разработанные диаграммы поведения обеспечивают комплексное представление о процессах, взаимодействиях и состояниях в системе, что является необходимым для дальнейшего анализа и оптимизации функциональности платформы.

1. **Ответы на вопросы**

1. Укажите виды диаграмм поведения. Какая между ними связь?

Диаграммы поведения в UML включают диаграмму деятельности, диаграмму состояний, диаграмму последовательностей, диаграмму коммуникации, диаграмму временных характеристик и диаграмму взаимодействия. Эти диаграммы помогают моделировать динамическое поведение системы, показывая, как объекты взаимодействуют друг с другом и как система реагирует на различные события. Связь между ними заключается в том, что каждая диаграмма акцентирует внимание на различных аспектах поведения системы: диаграммы деятельности и состояний описывают внутреннюю динамику, в то время как диаграммы последовательностей и коммуникации фокусируются на взаимодействии между объектами.

2. Опишите назначение диаграммы деятельности.

Диаграмма деятельности используется для моделирования последовательности действий или потоков управления в процессе. Она позволяет визуализировать рабочие процессы, алгоритмы и бизнес-процессы, отображая логическую последовательность шагов, включая разветвления, циклы и параллельные действия.

3. Опишите основные нотации, которые используются на диаграмме состояний.

На диаграмме состояний используются следующие нотации: состояния, представляющие различные этапы или режимы объекта; переходы, которые показывают возможные изменения состояния; начальные и конечные состояния, указывающие начало и завершение жизненного цикла; а также композитные состояния и подсостояния, описывающие более сложные состояния, включающие вложенные состояния и внутренние переходы.

4. Укажите виды связей между объектами на диаграмме последовательностей.

На диаграмме последовательностей связи между объектами представлены с помощью сообщений. Основные типы связей включают синхронные вызовы, асинхронные вызовы и ответы. Также можно использовать возвраты, сообщения создания и уничтожения объектов для моделирования инициации и завершения взаимодействий.

1. Какая диаграмма позволяет моделировать параллельные вычисления?

Диаграмма деятельности наиболее подходит для моделирования параллельных вычислений, так как поддерживает разветвления и параллельные потоки.